



PATENT  
Attorney Docket No.: 16869B-098400US  
Client Ref. No.: HAL300  
(340301717US01)

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of:

YOSHIKI KANO

Application No.: 10/812,537

Filed: March 29, 2004

For: METHOD AND APPARATUS  
FOR MULTISTAGE VOLUME  
LOCKING

Customer No.: 20350

Examiner: Unassigned

Technology Center/Art Unit: 2186

Confirmation No.: 3431

**PETITION TO MAKE SPECIAL FOR  
NEW APPLICATION UNDER M.P.E.P.  
§ 708.02, VIII & 37 C.F.R. § 1.102(d)**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

This is a petition to make special the above-identified application under MPEP § 708.02, VIII & 37 C.F.R. § 1.102(d). The application has not received any examination by an Examiner.

(a) The Commissioner is authorized to charge the petition fee of \$130 under 37 C.F.R. § 1.17(i) and any other fees associated with this paper to Deposit Account 20-1430.

(b) All the claims are believed to be directed to a single invention. If the Office determines that all the claims presented are not obviously directed to a single invention, then Applicants will make an election without traverse as a prerequisite to the grant of special status.

09/14/2005 EAYALEW1 00000044 201430 10812537  
02 FC:1464 130.00 DA

(c) Pre-examination searches were made of U.S. issued patents, including a classification search and a foreign patent database search. The searches were performed on or around June 17, 2005, and were conducted by a professional search firm, Mattingly, Stanger Malur & Brundidge, P.C. The classification search covered Class 707 (subclasses 1, 8, 9, 10, and 100), Class 709 (subclasses 217, 225, 226, and 229), Class 710 (subclass 200), Class 711 (subclasses 100, 114, 148, and 152), Class 713 (subclasses 200 and 201), and Class 714 (subclass 6). Because of the large size of these subclasses, keywords were used to narrow of number of documents returned. The foreign patent database search was conducted using Espacenet database and Japanese patent database.

(d) The following references, copies of which are attached herewith, are deemed most closely related to the subject matter encompassed by the claims:

- (1) U.S. Patent No. 5,551,046;
- (2) U.S. Patent No. 5,832,484;
- (3) U.S. Patent No. 5,933,824;
- (4) U.S. Patent No. 6,151,659;
- (5) U.S. Patent No. 6,268,850 B1;
- (6) U.S. Patent No. 6,499,058 B1;
- (7) U.S. Patent Publication No. 2003/0182285 A1;
- (8) U.S. Patent Publication No. 2003/0187860 A1;and ✓
- (9) Japanese Patent Publication No. JP 2000-148714. ✓

(e) Set forth below is a detailed discussion of references which points out with particularity how the claimed subject matter is distinguishable over the references.

A. Claimed Embodiments of the Present Invention

The claimed embodiments relate to management of a storage system having a plurality of storage volumes and, more particularly, to managing a virtualized storage

subsystem in such a way that the attributes of both virtual and internal volumes may be managed on the virtualized storage subsystem.

Independent claim 1 recites a first storage subsystem to be coupled to an external device. The first storage subsystem comprises a first storage volume configured by at least one of first storage devices in the first storage subsystem; a second storage volume configured by at least one of second storage devices in a second storage subsystem coupled to the first storage subsystem; and a controller configured to manage the first storage volume and the second storage volume as a virtual volume. The controller issues a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state.

Independent claim 5 recites a method for managing a storage system, comprising presenting a plurality of storage volumes to a host via a first storage subsystem, the plurality of storage volumes including a first storage volume that maps to a storage area within the first storage subsystem and a second storage volume that maps to a storage area within a second storage subsystem that is different from the first storage subsystem; receiving at the first storage subsystem a first request from the host to modify an attribute of a target storage volume, the target storage volume being one of the plurality of storage volumes presented to the host; and sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume.

Independent claim 16 recites a computer readable medium including a computer program for managing a storage subsystem. The computer program comprises code for presenting a plurality of storage volumes to a host via a first storage subsystem, the plurality of storage volumes including a first storage volume that maps to a storage area within the first storage subsystem and a second storage volume that maps to a storage area within a second storage subsystem that is different from the first storage subsystem; code for receiving at the first storage subsystem a first request from the host to modify an attribute of a target storage volume, the target storage volume being one of the plurality of storage volumes presented to the host; and code for sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume.

Independent claim 18 recites a first storage subsystem coupled to a second storage subsystem, which stores data written by a host device, the second storage subsystem presenting at least one storage volume as a storage resource to the first storage subsystem, the first storage subsystem presenting at least one virtual volume as a storage resource to the host device. The first storage subsystem comprises a first storage volume configured by at least one of first storage devices in the first storage subsystem; and a controller being configured to manage the first storage volume and a second storage volume configured by at least one of second storage devices in the second storage subsystem as the at least one virtual volume; wherein the controller issues a lock request to the second storage subsystem when the controller receives a request from the host device to change an attribute of the second storage volume to write protect state.

One of the benefits that may be derived is that attributes of the storage resources of another storage subsystem can be controlled in a storage virtualization scenario.

B. Discussion of the References

1. U.S. Patent No. 5,551,046

The patent to Mohan et al., US 5551046, discloses a method for non-hierarchical lock management in a multi-system shared data environment. In a combination of multiple concurrently-executing database management systems which share data storage resources, efficient lock processing for shared data is implemented by hiding from a global lock manager the distinction between transaction-interest and cache-interest locks that are processed at the DBMS level. The local lock manager of a DBMS, in response to a request for either type of lock, may issue a request to the global lock manager for a system-level lock without disclosing to the global lock manager the type of lock requested of the local lock manager. After receipt of the system level lock, the local lock manager can grant either transaction or cache interest locks locally on a data resource if the combined mode of locally-held locks on that data resource is greater than or equal to the requested mode.

Each lock manager maintains a lock table. FIG. 3 illustrates a global lock manager lock table 70 and local lock manager tables 80 and 81 for local lock managers 46 and 47, respectively. Each lock table includes multi-field entries which assist the owning lock manager in processing, maintaining, and releasing locks. The global table 70 contains

entries for LP locks, one such entry being indicated by 72. Each global table entry includes a Name field which identifies the data resource to which the lock applies. For example, an LP lock on page PN has an entry in the Name field which includes the page identifier PN. The Name field may also include other information necessary to the design and operation of an implementing system. See, e.g., Abstract and column 5, lines 40-58.

Mohan et al. is directed to lock processing for shared data. While Mohan et al. discloses the use of lock table, it does not teach a lock request to change an attribute of another storage volume. More specifically, Mohan et al. fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

2. U.S. Patent No. 5,832,484

The patent to Sankaran et al., US 5832484, discloses a database system and methods for improving scalability of multi-user database systems by improving management of locks used in the system. The system provides multiple server engines, with each engine having a Parallel Lock Manager. More particularly, the Lock Manager decomposes the single spin lock traditionally employed to protect shared, global Lock Manager structures into multiple spin locks, each protecting individual hash buckets or groups of hash buckets which index into particular members of those structures. In this manner, contention for shared, global Lock Manager data structures is reduced, thereby improving the system's scalability. Further, improved "deadlock" searching methodology is provided. Specifically, the system provides a "deferred" mode of deadlock detection. Here, a task simply goes to sleep on a lock; it does not initiate a deadlock search. At a later point in time, the task is awakened to carry out the deadlock search. Often, however, a task can be awakened with the requested lock being granted. In this manner, the "deferred" mode of deadlock detection allows the system to avoid deadlock detection for locks which are soon granted.

Two types of locks are provided: an exclusive address lock and a shared address lock. The shared address lock is provided so that multiple readers of an object can do so in a shared fashion, thereby avoiding the need for the readers to serialize one after another. In one embodiment, the lock record includes a previous pointer (lrprev) and a next pointer (lrnext), for establishing a linked list of lock records--the "context chain". The context chain exists on a per task basis: given a task, the chain indicates the locks which the task holds. The next set of pointers, lrprev and lrnext, link the lock record to the log manager data structures. A spin lock for the hash bucket is provided by lrspinlock. This is a new data member which is added for supporting parallel lock management. An identifier representing the task holding or waiting for the lock is stored by lrspid. The next data member, lrtype, stores a lock type for logical locks. See, e.g., Abstract and column 27, lines 24-29; and column 27, line 57 to column 28, line 3.

Sankaran et al. is directed to improving management of locks in a multi-user database system. While Sankaran et al. discloses management of locks, it does not teach a lock request to change an attribute of another storage volume. More specifically, Sankaran et al. fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

3. U.S. Patent No. 5,933,824

The patent to DeKoning et al., US 5933824, discloses methods and associated apparatus for coordinating file lock requests from a cluster of attached host computer systems within I/O controllers (e.g., intelligent I/O adapters) attached to a storage subsystem. The I/O controllers, operable in accordance with the methods of the invention, includes semaphore tables used to provide temporary exclusive access to an identified portion of an identified file. The host systems request the temporary exclusive access of a file through the I/O controllers rather than over slower network communication media and protocols as is known in the art. The I/O controllers then manage a plurality of competing lock requests to provide mutual exclusivity of the file access. The file lock management is therefore managed over the higher

bandwidth storage interface channels of the host systems and without the generalized network protocols burdening the lock management process and the host system CPUs. The I/O controllers in which the methods of the invention are operable, as referred to herein, includes the controller within a storage device such as a RAID subsystem and decentralized control storage devices such as a RAID subsystem or a storage subsystem with control decentralized to a plurality of intelligent host adapters associated with the cluster of host systems.

The controller determines if other previously granted locks are for overlapping portions of the same file. Each lock that is granted is stored in a table entry retaining the file identifier and the associated extent of the granted locks along with an allocated semaphore used to lock the identified file. The method searches the table of granted locks to determine if a new lock request overlaps the locked portions of granted file locks. If a newly requested file lock overlaps a previously granted file lock, the new file lock request must await release of the granted lock. The request is deferred until the overlapping lock(s) are released. If no overlapping locks are located, the newly requested file lock may be granted immediately. See, e.g., Abstract and column 7, lines 51-67.

DeKoning et al. is directed to coordinating lock requests from a cluster of host computer systems. While DeKoning et al. discloses coordinating lock requests, it does not teach a lock request to change an attribute of another storage volume. More specifically, DeKoning et al. fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

4. U.S. Patent No. 6,151,659

The patent to Solomon et al., US 6151659, discloses a distributed raid storage system that has at least three data storage disks and a plurality of processing nodes in communication with the data storage disks. Each of the processing nodes shares access to the data storage disks, and each of the processing nodes includes a distributed lock manager that allows or denies access to selected stripes of data storage sectors on any of the data storage

disks. Each of the processing nodes includes an interface to a private communication link to a single one of a plurality of host operating systems.

One embodiment employs a distributed lock manager 102 which runs on each processing node. This manager is responsible for arbitrating access to the data stripes. The manager ensures there is no contention between processors for access to different portions of the same disk drive, and if there is a conflict, the manager coordinates the access so as to insure consistency of the parity data with the write data. In a preferred embodiment, the management is achieved through setting and releasing locks of various types depending on the disk access desired. See, e.g., Abstract and column 4, lines 3-14.

Solomon et al. is directed to a distributed lock manager. While Solomon et al. discloses the setting and releasing of locks, it does not teach a lock request to change an attribute of another storage volume. More specifically, Solomon et al. fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

5. U.S. Patent No. 6,268,850 B1

The patent to Ng, US 6268850, discloses a user interface that permits a programmer or other person to manage lock groups for classes. The programmer enters information through the user interface to define new lock groups, update defined lock groups, and delete lock groups. The programmer manages the lock groups in the classes, and an optional mapping tool maps the defined lock groups when converting data between an object model and a relational model.

The user interface may be used to modify or delete existing lock groups. A mapping tool can map the lock groups during mapping of data between objects in an object-oriented model and tables in a relational model. Thus, a user's specified lock groups are saved and need not be repeatedly re-created. See, e.g., Abstract and column 3, lines 34-40.



Ng is directed to managing lock groups for classes. While Ng discloses the use of lock groups, it does not teach a lock request to change an attribute of another storage volume. More specifically, Ng fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

6. U.S. Patent No. 6,499,058 B1

The patent to Hozumi, US 6499058, discloses a conventional data shared system using a plurality of processing nodes and data storage units in a storage area network using SAN OS was a volume-level locking or a file-system-level locking through one limited server. A locking system for SAN proposed this time is one that is a file-system-level locking and creates no single point of failure. Namely, the locking system is incorporated into each storage unit of Storage Area Network to run software. As a result, the storage unit is converted to an intelligent form and an acceptor (1) for a first protocol and an acceptor (2) for a second protocol coexist. This allows the acceptor (1) to perform a locking mechanism and the acceptor (2) to perform data transfer, so that the locking system that is a file system level locking and that creates no single point of failure can be realized. The plurality of protocols is thus used so as to execute data control and data transfer efficiently.

The lock mechanism relates to a method for sufficiently bringing about high-speed performance of storage area network without causing single-point-of-failure and relates to its apparatus. An apparatus having embedded software is used, and is put in a storage unit in SAN and manages a file system of only data stored in the storage unit. The use of such an apparatus makes it possible to manage the lock, which is conventionally managed by one processing unit, in a spread manner on a storage side corresponding to each data. See, e.g., Abstract and column 6, lines 19-28.

Hozumi is directed to a locking mechanism for file system level locking. While Hozumi discloses managing a lock to create no single point of failure, it does not teach

a lock request to change an attribute of another storage volume. More specifically, Hozumi fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

7. U.S. Patent Publication No. 2003/0182285 A1

The published patent application to Kuwata, 20030182285, discloses a file locking method and implementation that allows a plurality of user sessions to open and read a file, but at any one time, only one session will be allowed to change the data displayed in the browser window and to update the file. This file locking method sets up a file access priority by using file locks that are date-time stamped and session stamped. The types of lock associated with the present invention include read lock, authority lock, write lock, and folder lock. When a session/user requests access to a file, the application will check a lock table associated with the requested file. For each lock on the file, there is an entry in the lock table for each of the attributes of the lock: lock type, session owner, date-time stamp. Depending on the lock and the existing locks on the file, the requesting session may be granted a lock. After the access request is fulfilled, the file lock may be removed. When a session expires, all the locks owned by this session will be invalidated and removed.

The file locking method works by enabling a file lock depending on the priority of the types of lock and the priority of the lock request. Various realizations of the invention involve the operation of the lock system with respect to three common types of requests: read, write and modify. These operations can be applied to both documents and folders. The locks are preferably implemented as temporary files with the attributes of the locks encoded in the file name to facilitate searching and comparing lock types. Each lock file name contains the name of the lock item, combined with the requested date-time in milliseconds, the session ID and the type of lock. See, e.g., Abstract and paragraphs [0024]-[0025].

Kuwata et al. is directed to a file locking for setting up a file access priority by using file locks that are date-time stamped and session stamped. While Kuwata et al. discloses the use of file locks with attributes encoded in the file name, it does not teach a lock request to change an attribute of another storage volume. More specifically, Kuwata et al. fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

8. U.S. Patent Publication No. 2003/0187860 A1

The published patent application to Holland, 20030187860, discloses using whole-file and dual-mode locks to reduce locking traffic in data storage systems. This is a methodology wherein two different types of locks are used by a storage manager when multiple clients wish to access a particular redundantly-stored file. Simple byte-range based mutual exclusion (or mutex) locks are granted by the storage manager for data writes/updates to the file when the file is in the fault-free state, and individual readers/writers (R/W) locks are granted by the storage manager when the file is in the degraded state. No read locks are required of clients when the file object is in the fault-free state. During the fault-free state of the file object, when exactly one client is writing to the file object, the storage manager grants that file object a whole-file lock valid over the entire file object. Each client may have a client lock manager that interacts with appropriate storage manager lock manager to request and obtain necessary locks. These various locking mechanisms reduce lock-related network traffic in a data storage system.

In one embodiment, the lock table maintained by the CLM may contain a record of the identity of the CAP (the CAP# column in FIG. 3) that has been granted a lock, the object ID (the object # column in FIG. 3) of the object being accessed by the CAP, and the byte-range over which the lock has been granted to the CAP. Some additional information in the CLM lock table may include, per entry, an indication whether the corresponding lock is held active or inactive and whether the lock is for a data read operation

or a data write operation. Generally speaking, the lock table in the CLM 42 is specific to the CAPs running on the respective client computer. See, e.g., Abstract and paragraph [0046].

Holland relates to the use of two different types of locks by a storage manager: simple byte-range based mutual exclusion locks when the file is in the fault-free state (see FIG. 4) and individual readers/writes locks when the file is in the degraded state (see FIG. 5). It does not teach a lock request to change an attribute of another storage volume. More specifically, Holland fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

9. Japanese Patent Publication No. JP 2000-148714

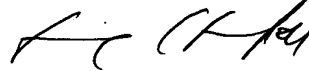
The Patent to Nakajima, JP 2000148714, discloses lock management between systems that uses a coupling mechanism without using a communication means other than the list structure of the coupling mechanism managing lock information by equipping a global lock manager with a communication function between respective local lock managers. A lock table comprises the global lock manager and local lock managers. To a lock request from a coupling mechanism lock manager, an option indicator is added. At the lock request, the lock table is compared and possibly updated. When the lock request is successful, the lock table is updated and lock information is registered in a list for lock information storage specified with a holder ID. If the lock request ends in failure and the option indicator is requested, the list number of a list for communication is found from a lock conflict control table 240 and lock information is registered in the list for communication.

Nakajima et al. is directed to managing lock information by a global lock manager. While Nakajima et al. discloses the use of a lock table and a lock request, it does not teach a lock request to change an attribute of another storage volume. More specifically, Nakajima et al. fails to teach issuing a lock request to the second storage subsystem when the controller receives a request from the external device to change an attribute of the second storage volume to write protect state, as recited in independent claims 1 and 18; or sending a

second request from the first storage subsystem to the second storage subsystem if the target volume is determined to be the second storage volume, the second request being a request to modify the attribute of the target volume, as recited in independent claims 5 and 16.

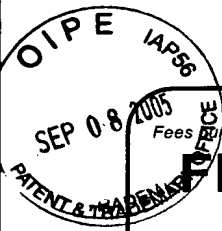
(f) In view of this petition, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,



Chun-Pok Leung  
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, 8<sup>th</sup> Floor  
San Francisco, California 94111-3834  
Tel: 650-326-2400  
Fax: 415-576-0300  
Attachments  
RL:rl  
60579508 v1



Effective on 12/08/2004.

Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

# FEE TRANSMITTAL

## For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27**TOTAL AMOUNT OF PAYMENT** (\$)**330.00****Complete if Known**

Application Number	10/812,537
Filing Date	March 29, 2004
First Named Inventor	Kano, Yoshiki
Examiner Name	Unassigned
Art Unit	2186
Attorney Docket No.	16869B-098400US

**METHOD OF PAYMENT** (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): \_\_\_\_\_  
☒ Deposit Account Deposit Account Number: 20-1430 Deposit Account Name: Townsend and Townsend and Crew LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments**WARNING:** Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038**FEE CALCULATION****1. BASIC FILING, SEARCH, AND EXAMINATION FEES**

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Small Entity	Fee (\$)	Small Entity	Fee (\$)	Small Entity	Fee (\$)	
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

**2. EXCESS CLAIM FEES**

Fee Description	Small Entity	
	Fee (\$)	Fee (\$)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee (\$)	Fee Paid (\$)	Multiple Dependent Claims	Fee (\$)	Fee Paid (\$)
18	-20 or HP = 0	x \$50	= \$0			
Indep. Claims	Extra Claims	Fee (\$)	Fee Paid (\$)			
4	-3 or HP = 1	x \$200	= \$200			

HP = highest number of independent claims paid for, if greater than 3

**3. APPLICATION SIZE FEE**

If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee (\$)	Fee Paid (\$)
- 100 =	/ 50 =	(round up to a whole number) x		

**4. OTHER FEE(S)**

Non-English Specification, \$130 fee (no small entity discount)

Other: PETITION TO MAKE SPECIAL**130.00****SUBMITTED BY**

Signature		Registration No. (Attorney/Agent) 41,405	Telephone 650-326-2400
Name (Print/Type)	Chun-Pok Leung		Date September 8, 2005

# DATA PROCESSING SYSTEM

**Patent number:**

JP2000148714

**Publication date:**

2000-05-30

**Inventor:**

NAKAJIMA TAKAO; UCHIDA TOMONARI; YOKOTA HIROSHI; TAKAYAMA YOSHITO

**Applicant:**

HITACHI LTD

**Classification:**

- international:

G06F15/177

- european:

**Application number:**

JP19980318699 19981110

**Priority number(s):**

JP19980318699 19981110

**Abstract of JP2000148714**

**PROBLEM TO BE SOLVED:** To actualize lock management between systems which uses a coupling mechanism without using a communication means other than the list structure of the coupling mechanism managing lock information by equipping a global lock manager with a communication function between respective local lock managers.

**SOLUTION:** A lock table 200 comprises the global lock manager and local lock managers. To a lock request 300 from a coupling mechanism lock manager, an option indicator 310 is added. At the lock request, the lock table 200 is compared and possibly updated. When the lock request is successful, the lock table 200 is updated and lock information is registered in a list 211 for lock information storage specified with a holder ID. If the lock request ends in failure and the option indicator 310 is requested, the list number of a list 212 for communication is found from a lock conflict control table 240 and lock information is registered in the list 212 for communication.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-148714

(P2000-148714A)

(43) 公開日 平成12年5月30日 (2000.5.30)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 15/177

識別記号

6 8 2

F I

C 0 6 F 15/177

テーマコード(参考)

6 8 2 F 5 B 0 4 6

審査請求 未請求 請求項の数2 O L (全11頁)

(21) 出願番号 特願平10-318699

(22) 出願日 平成10年11月10日 (1998.11.10)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 中島 隆夫

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部内

(72) 発明者 内田 智斉

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部内

(74) 代理人 100068504

弁理士 小川 勝男

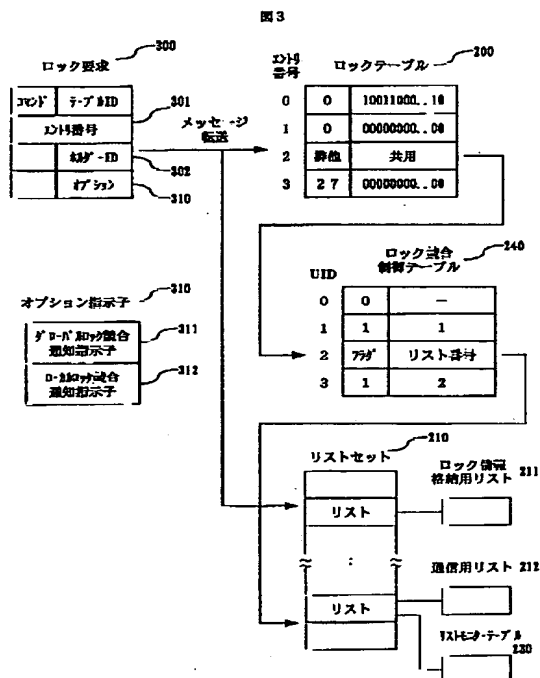
最終頁に続く

(54) 【発明の名称】 データ処理システム

(57) 【要約】

【課題】ロック情報を管理する結合機構のリスト構造以外の通信手段を用いることなく、結合機構を使用したシステム間のロック管理を実現すること。

【解決手段】ロック競合時の通信用リストと、ロック競合時の通知先を登録するロック競合制御テーブルを備え、ロックが競合してロック要求が失敗した場合に、通信用リストにロック要求を書き込み、通信先のシステムに受信データの到着を通知する。





## 【特許請求の範囲】

【請求項1】1台以上のプロセッサ上の各プロセッサで動作する複数のデータベースマネージャと、前記各々のデータベースマネージャが持つローカルロックマネージャと、前記各々のローカルロックマネージャと接続されるグローバルロックマネージャを持った結合機構とを含んで構成され、前記データベースマネージャによって管理される共用データに対する、多数のトランザクションによるアクセスを管理するロック機構を保持するデータ処理システムにおいて、前記グローバルロックマネージャは前記各々のローカルロックマネージャ相互間の通信機能を備えていることを特徴とするデータ処理システム。

【請求項2】前記グローバルロックマネージャは、ロック情報を管理するロックテーブル及びリストと、ローカルロックマネージャ相互間の通信のために使用するリストと、ロック競合が発生した場合にリストエントリを登録するリストの番号を登録するロック競合制御テーブルを含んだリスト構造を持ち、ロック競合制御テーブルの各エントリに対応するリストの番号を登録するコマンドを持ち、各プロセッサ上のローカルロックマネージャからのロックテーブルエントリの更新を行うコマンドは、当該コマンドの実行時にロックテーブルエントリの比較に失敗した場合に、ロック競合制御テーブルに登録されているリストにリストエントリを登録するか否かを選択する要求指示子を持ち、当該コマンドの実行時にロックテーブルエントリの比較に失敗し、前記要求指示子の指定がある場合に、ロック競合制御テーブルに登録されているリストにリストエントリを登録し、当該リストにリストモニターが登録されていてかつリスト状態が遷移した場合に、リスト通知を行うことを特徴とする請求項1記載のデータ処理システム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明はデータ処理システムに係り、特に複数のシステム間で共有する資源のロック管理に関する。

## 【0002】

【従来の技術】複数の処理装置（中央処理複合体：CEC）と、これらの処理装置により共用される結合機構（構造化電子記憶機構：SES）と、それらを結合するメッセージバスとからなるデータ処理システムに関する従来技術として、例えば、特開平6-83783号公報に記載された技術が知られている。

【0003】特開平6-83783号公報に記載の従来技術は、処理装置間のデータ排他制御、処理装置間のメッセージ通信機能を持つ結合機構のリスト構造に関するものである。

【0004】また、結合機構を用いて共用資源のロック管理を行うデータ処理システムに関する従来技術とし

て、例えば、米国特許第860,808号公報に記載された技術が知られている。

【0005】米国特許第860,808号公報に記載の従来技術は、複数のMVSシステムによって構成されるシスプレックス環境で動作し、TLM（tailored lock manager：データベースマネージャ固有のロック管理を行う）と、SLM（system lock manager：システム共通のロック管理を行う）から構成されるローカルロックマネージャを持ったデータベースマネージャと、結合機構上のリスト構造上でロック情報を管理するSESLM（SESロックマネージャ）と、各システム上のSLM間の通信手段によって実現される結合機構を用いた複数システム間のロック管理に関するものである。

【0006】ロック要求は最初に各システム上のローカルハッシュテーブルに基づくローカルレベルで処理され、可能であればこのレベルで共有資源へのアクセスが許可または拒否される。必要があればSLMがSESLMと通信を行い、ロックテーブルとリストを含んだ結合機構上のリスト構造内のロック情報を更新する。

【0007】あるシステムが排他的にロック確保を要求した場合に、別のシステムが既に共用でロックを保持している場合、SLMからSESLMへのロック要求が失敗することによってロック競合を検出する。排他的にロックを確保しようとしているシステムから、他のシステムに対して、前述のSLM間の通信手段を用いて当該資源がグローバル管理下に置かれる旨を通知して、各システムのローカルレベルのロック情報を取得することにより、排他的にロックを要求しているシステムによってロック情報がグローバルレベルで管理される。

## 【0008】

【発明が解決しようとする課題】前述した従来技術は、ロック情報を管理する結合機構のリスト構造とは別に、ロック競合が発生した場合に使用する各システムのSLM間の通信手段を別に備えていなければならない。また、ロック競合が生じた場合は、SLMからSESLMへのロック要求が失敗することにより、ロック競合が発生したことを検知した後、前記SLM間の通信手段を介して、他のシステムのSLMへ競合の発生を通知しているが、この通知に伴うメッセージの送信処理はSLMの数すなわちシステムの数に比例して増大する。

【0009】本発明の目的は、ロック情報を管理する結合機構のリスト構造以外に別の通信手段を用意することなく、結合機構を使用したシステム間のロック管理を実現することにある。

【0010】また、本発明のもう1つの目的は、ロック競合の発生に伴い他のシステムのSLMへロック競合の発生を通知した場合に、ロック競合を検知したシステムからのメッセージ送信処理を削減し、SLMの数すなわちシステムの数が増加した場合のオーバーヘッドを軽減す

ることにある。

#### 【0011】

【課題を解決するための手段】本発明によれば、前記目的は、SESLM内にメッセージ通信用リストを備え、各システムのSLM間のメッセージ通信をSESLMを経由して行うことにより達成される。さらに、SESLM内にロック競合時の通信先を登録するロック競合制御テーブルを備え、ロック競合が発生した場合に、ロック要求を行ったSLMにロック競合の発生を応答すると同時に、他のシステムのSLMに対する通信用リストにロック競合の発生を通知するメッセージを含んだリストエントリを書き込み、通信先のシステムに受信データの到着を通知することによって、ロック競合の発生を通知するメッセージの送信処理を行う。

【0012】これにより、ロック競合を引き起こしたロック要求を行ったシステムが、ロック競合の応答を受信した後に行っていた、他のシステムのSLMにロック競合を通知するメッセージの送信を不要にすることにより達成される。

#### 【0013】

【発明の実施の形態】次に、本発明について図面を参照して詳細に説明する。

【0014】図1は、本発明によるシステム構成図である。本発明によるシステム構成は、複数のシステム100-1、100-2、共用データベース120および結合機構110から構成される。システムは、共用データベースにアクセスする複数のアプリケーションプログラム(AP:101-1、101-2)、アプリケーションプログラムからのデータベースへのアクセスを制御するデータベースマネージャー(DBM:102-1、102-2)、データベース管理プログラムから共用データベースに対するロック要求を受け付けるテーラードロックマネージャー(TLM:103-1、103-2)、TLMからのロック要求によってシステム間でロックの制御を行うシステムロックマネージャー(SLM:104-1、104-2)を含む。

【0015】結合機構は、各システム上のロックマネージャーから要求されたロック要求および通信要求の制御を行う結合機構ロックマネージャー111を含む。

【0016】図2は、本発明の一実施例が適用された結合機構ロックマネージャーSESLMとシステムロックマネージャーSLMのテーブル構造を示した図である。結合機構ロックマネージャーは、グローバルロックマネージャー(GLM, 201)とローカルロックマネージャー(LLM, 202)から構成される既存のロックテーブル200と、リストセット210に、ロック競合制御テーブル240が追加される。また、リストセットは既存のロック情報格納用リスト211に、通信用リスト212が追加される。リストはリストエントリ220から構成される。また、送信用リストはリストモニターテ

ーブル230を含む。

【0017】ロック競合制御テーブルは各エントリが有効かどうかを示すフラグ241と通信用リストのリスト番号242を含む。TLM250からSLM260へはSLMパラメタリスト251によってロック要求が行われる。SLMではロック要求をローカルハッシュテーブル261またはグローバルハッシュテーブル262で監視する。

【0018】図3は、SLMからのロック要求と、結合機構ロックマネージャー内のテーブルの関連を示す図である。ロックマネージャーからのロック要求300には、オプション指示子310が追加される。オプション指示子にはグローバルロック競合通知指示子311とローカルロック競合指示子312を含む。ロック要求によってロックテーブル200が比較され、更新されることがある。

【0019】ロック要求が成功すれば、ロックテーブルが更新され、ホルダーIDで指定されるロック情報格納用リスト211にロック情報が登録される。ロック要求が失敗すれば、ロックテーブルは更新される場合と更新されない場合があり、オプション指示子の要求があれば、ロック競合制御テーブル240から通信用リスト212のリスト番号を求めて、通信用リストにロック情報が登録される。

【0020】図4は、SESLMのロック要求の処理手順を示す図である。最初にロック要求の妥当性をチェックし(処理401、402)、不当な要求の場合は要求例外を示す応答コードでリターンする処理403。次にロックテーブル処理を実行する処理404。ロックテーブル処理では、米国特許第860、808号公報に記載されているのと同様の処理が行われ、ロック要求中のエントリ番号によって示されるロックテーブルエントリに対して比較と置き換えが行われる。

【0021】GLMがゼロでない場合はグローバルロックマネージャー比較が失敗する。GLMがゼロの場合は、GLMにホルダーIDの値を書き込み、LLMが比較される。LLMはビットストリングであり、最左を0としてホルダーIDの値に相当するビット位置以外の全てのビットを比較し、いずれか1ビットでもゼロでない場合はローカルロックマネージャー比較が失敗し、全てがゼロの場合はローカルロックマネージャー比較が成功する。

【0022】グローバルロックマネージャー比較に失敗した場合については、図5で説明する(処理405)。ローカルロックマネージャー比較に失敗した場合については、図6で説明する(処理406)。グローバルロックマネージャー比較およびローカルロックマネージャー比較が成功した場合は、リストエントリの登録処理を行う。新規のリストエントリの登録の場合は、新規にリストエントリを生成し、指定されたデータを書き込む(処

理407、408)、既存のリストエントリが存在する場合は、指定されたデータにより更新する処理409。以上の処理が完了した後、正常終了の応答コードでリターンする(処理410)。

【0023】図5は、グローバルロックマネージャー比較に失敗した場合の処理を示す図である。グローバルロック競合通知要求311が指定されているかどうか判定する処理501。指定されていなければ、グローバルロックマネージャー比較失敗の応答コードでリターンする(処理510)。指定されている場合は、GLMの値に対応するロック競合制御テーブル240のエントリからリスト通信用リスト番号242を求める(処理502)。

【0024】次にリストエントリを生成するための空きエントリの有無を判定し(処理503)、次に送信用リストに空きがあるかどうかを判定する(処理505)。空きリストエントリがなければセカンダリ応答コードとしてリストセットフルを設定してリターンする処理504、510。送信用リストに空きがなければ、セカンダリ応答コードとしてリストフルを設定し、リターンする(処理506、510)。共に空きがあればリストエントリを生成し、指定されたデータを書き込む(処理507)。

【0025】通信先のシステムにリスト通知コマンドを送信するためのリストモニターが登録されているかどうかをリストモニターテーブル230を参照して判定し、登録されていればさらに登録されているリストエントリ数が1かどうかを判定する(処理508)。なぜならば、リスト通知コマンドはリストにリストエントリが登録されていない状態からリストエントリが登録された場合に発行するためである。登録されているリストエントリ数が1の場合はリスト通知コマンドを発行する(処理509)。以上の処理が完了した後、グローバルロックマネージャー比較失敗の応答コードでリターンする(処理510)。

【0026】図6は、ローカルロックマネージャー比較に失敗した場合の処理を示す図である。ローカルロック競合通知要求312が指定されているかどうか判定する(処理601)。指定されていなければ、ローカルロックマネージャー比較失敗の応答コードでリターンする(処理611)。指定されている場合は、ロック競合制御テーブル240に登録されている有効エントリのうち、ホルダーID302で示すエントリ番号を除いた全てのエントリのリスト番号242を求める(処理602)。

【0027】次に全てのリストエントリを生成するための空きエントリの有無を判定し(処理603)、次に各送信用リストに空きがあるかどうかを判定する(処理605)。空きリストエントリがなければセカンダリ応答コードとしてリストセットフルを設定してリターンする

(処理604、611)。送信用リストに空きがなければセカンダリ応答コードとしてリストフルを設定し、リターンする(処理606、611)。共に空きがあればリストエントリを生成し、指定されたデータを書き込む(処理607)。

【0028】通信先のシステムにリスト通知コマンドを送信するリストモニターが登録されているかどうかをリストモニターテーブル230を参照して判定し、登録されていればさらに登録されているリストエントリ数が1かどうかを判定する(処理608)。登録されているリストエントリ数が1の場合はリスト通知コマンドを発行する(処理609)。全てのリストエントリの処理が完了していなければ処理607から繰り返す(処理610)。以上の処理が完了した後、グローバルロックマネージャー比較失敗の応答コードでリターンする(処理611)。

【0029】図7は、SLMによるSESLMの初期化処理手順を示す図である。以下の説明中の既存のコマンドについては、特開平6-83783号公報を参照されたい。最初にALSTコマンドによりSESLM用のリスト構造の割り当てを行う(処理701)。ALSTコマンドには、既存のロックテーブル200、リストセット210に加えて、ロック競合制御テーブル240を生成するためにLSTオペランドのサブオペランドとしてLCCIオペランドが新設され、処理701ではLCCIオペランドを指定する。

【0030】次に、通信を行うための準備として、ALSUコマンドによりユーザー登録を行い(処理702)、RLMコマンドによりリストモニター登録を行い(処理703)、受信用リストの初期化のためにリストが空になるまでRDLEコマンドを発効する(処理704)。さらに、新規コマンドであるロック競合制御テーブル書き込みWLCCコマンドにより、受信用リスト番号をロック競合制御テーブルへ登録する(処理705)。

【0031】図8は、SLMによるロック確保要求の処理手順を示す図である。SLMパラメタリスト251で示されるハッシュインデックスが使用中かどうか、ローカルハッシュテーブルをチェックする(処理801)。使用中であればグローバル管理中かどうかチェックする(処理802)。ローカル管理からグローバル管理へは処理812および図9及び図10のロックエスカレーションによって遷移する。グローバル管理中であれば、そのハッシュインデックスを管理しているSLM(グローバルマネージャーと呼ぶ)ヘシグナルを送信する(処理803)。

【0032】グローバル管理中でなければ当該ハッシュインデックスの管理状態と同じ要求であるかチェックする(処理804)。同じ要求SHR状態でSHR要求が行われた、またはEXCL状態の場合はローカルハッシ

ュテーブルに登録し(処理805)、TLMへリターンする(処理806)。処理801で未使用の場合と、処理804で異なる要求SHR状態でEXCL要求が行われた場合はSESLMのロックテーブルエントリを更新する(処理807)。結果を判定し(処理808)、更新が成功した場合はTLMへリターンする(処理809)。

【0033】更新が失敗した場合、競合が発生しているかどうかを応答コードによりチェックする(処理810)。競合が発生していなければ、不当な要求としてエラーリターンする(処理811)。競合が発生している場合は、ロックエスカレーション処理を行い(処理812)、ローカルハッシュテーブルに登録し(処理813)、TLMにリターンする(処理809)。

【0034】本特許の特徴であるロックエスカレーション処理については図9及び図10で詳細に説明する。その他の処理の詳細については、米国特許第860,808号公報を参照されたい。

【0035】図9および図10は、ロックエスカレーション処理を示すタイミングチャートである。最初にTLM1から資源名RNAMEを共用SHRで確保する要求が行われる901。SLM1はSESLMのロックテーブルエントリをSHR状態を書き込み902、TLM1へロック確保成功を応答する903。

【0036】次にTLM2から資源名RNAMEを排他EXCLで確保する要求が行われる904。SLM2はロックテーブルエントリへのEXCL状態の書き込みとレコードデータの書き込みを行う905。ロックテーブルエントリへの書き込みを行う時に、ローカルロック競合通知指示312を行う。ロックテーブルエントリはすでにSHR状態であるためローカルロックマネージャー比較に失敗し、SLM1への送信用リストにリストエントリが登録され、リスト通知が行われる。

【0037】これによりSLM2ではローカルロックマネージャー比較失敗を検知した後に従来行っていたSLM1へのエスカレーションシグナルの送信処理が省略される。SLM2は応答コードよりロック競合の発生とエスカレーション通知が行われたことを検知する906。SLM1ではリスト通知を受信し907、リストエントリを読み出してエスカレーションシグナルを受信し908、ローカルハッシュテーブルからロック情報を読み出してSLM2へSESLMを経由してエスカレーション応答を送信する909。SLM2はリスト通知を受信し

910、リストエントリを読み出してエスカレーション応答を受信し911、グローバルハッシュテーブルを構築し912、以降のロックエスカレーションに伴う処理を継続する913。

【0038】尚、上記以外の処理で、各SLM間の通信処理を行う場合についても、SESLMを経由してメッセージが送受信される。

【0039】

【発明の効果】以上説明したように本発明によれば、ロック情報を管理する結合機構のリスト構造以外の通信手段を用いることなく、結合機構を使用したシステム間のロック管理を実現することができる。

【0040】さらにロック競合が発生した場合に、ロック競合を引き起こしたロック要求を行ったシステムから他のシステムへロック競合の発生を通知するためのメッセージの送信が不要となるため、ロック競合に伴うメッセージの送信処理のオーバーヘッドが削減され、SLMの数すなわちシステムの数が増加した場合のオーバーヘッドが軽減される。

【図面の簡単な説明】

【図1】本発明によるデータ処理システムの構成図である。

【図2】SESLMとSLMのテーブル構造を示した図である。

【図3】SLMからのロック要求と、結合機構ロックマネージャー内のテーブルの関連を示す図である。

【図4】SESLMのロック要求の処理手順を示すフローチャートである。

【図5】グローバルロックマネージャー比較に失敗した場合の処理を示すフローチャートである。

【図6】ローカルロックマネージャー比較に失敗した場合の処理を示すフローチャートである。

【図7】SLMによるSESLMの初期化処理手順を示すフローチャートである。

【図8】SLMによるロック確保要求の処理手順を示すフローチャートである。

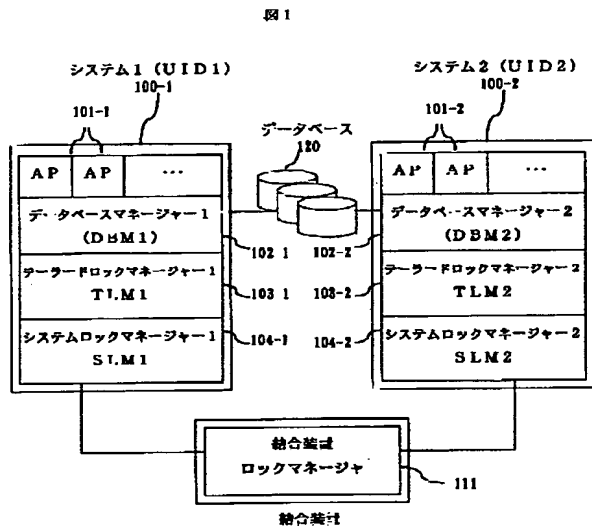
【図9】ロックエスカレーション処理を示すタイミングチャートである。(その1)

【図10】ロックエスカレーション処理を示すタイミングチャートである。(その2)

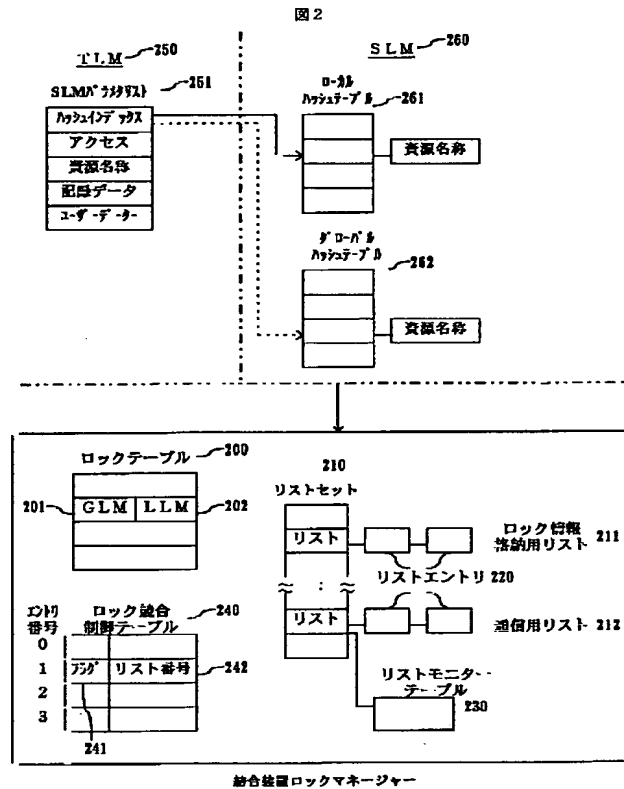
【符号の説明】

100…システム、 110…結合機構、 120…データベース。

【図1】

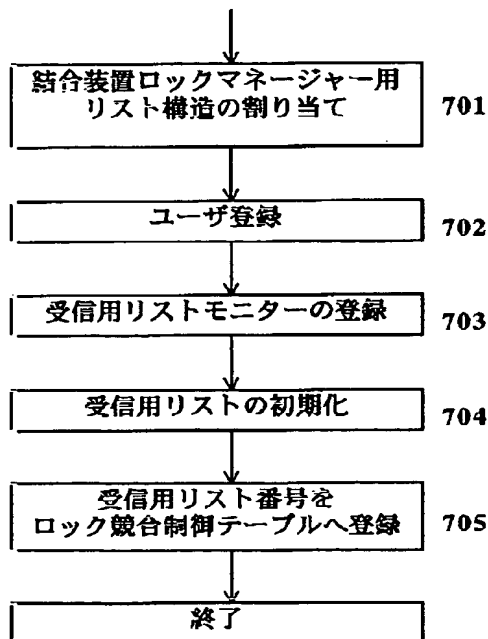


【図2】

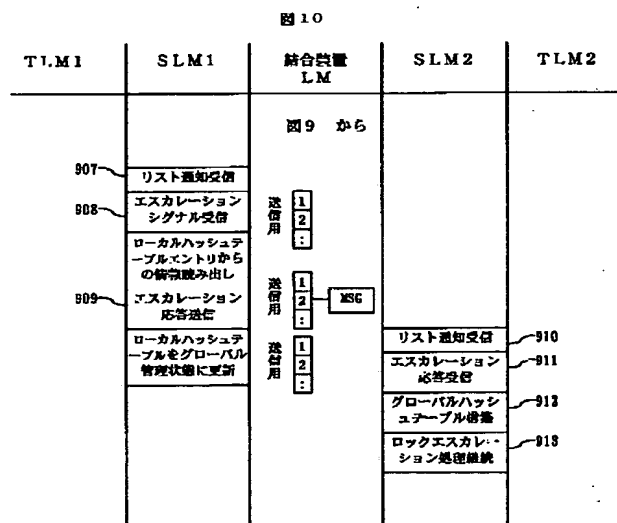


【図7】

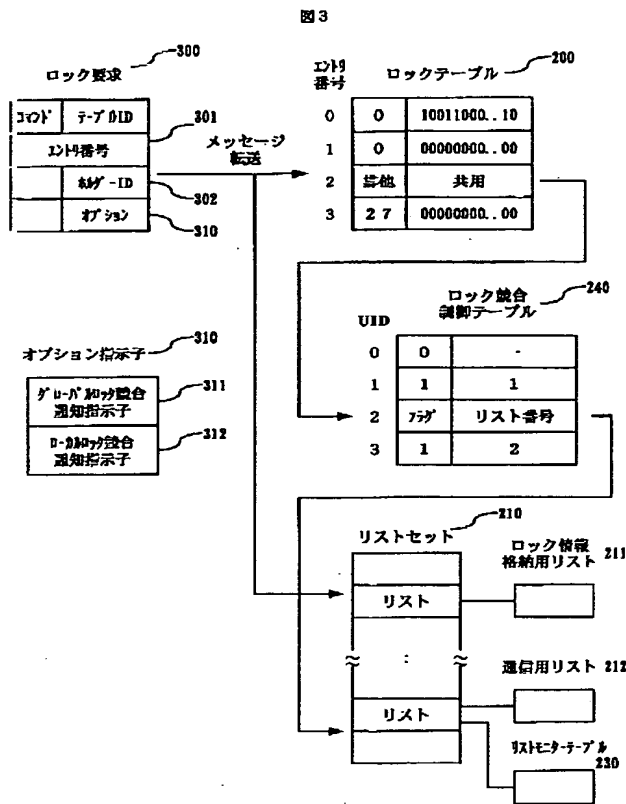
図7



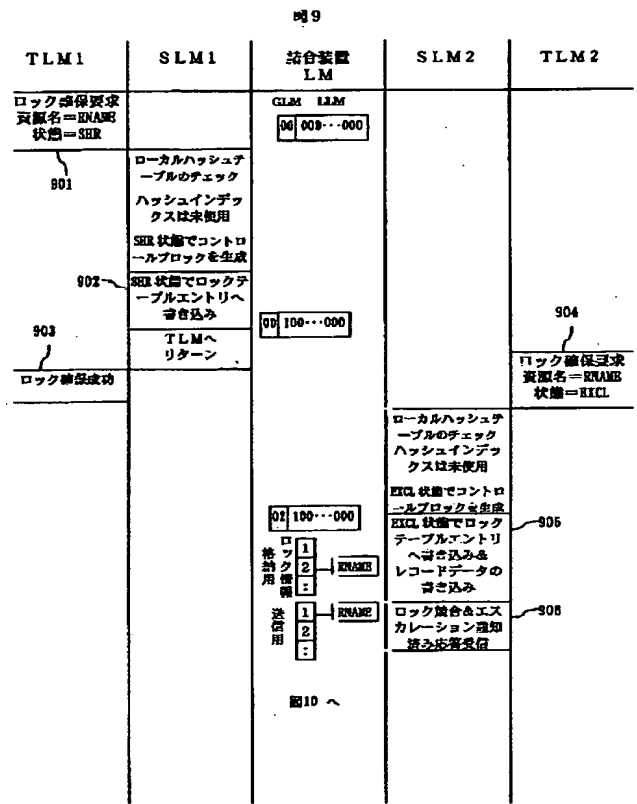
【図10】



【図3】

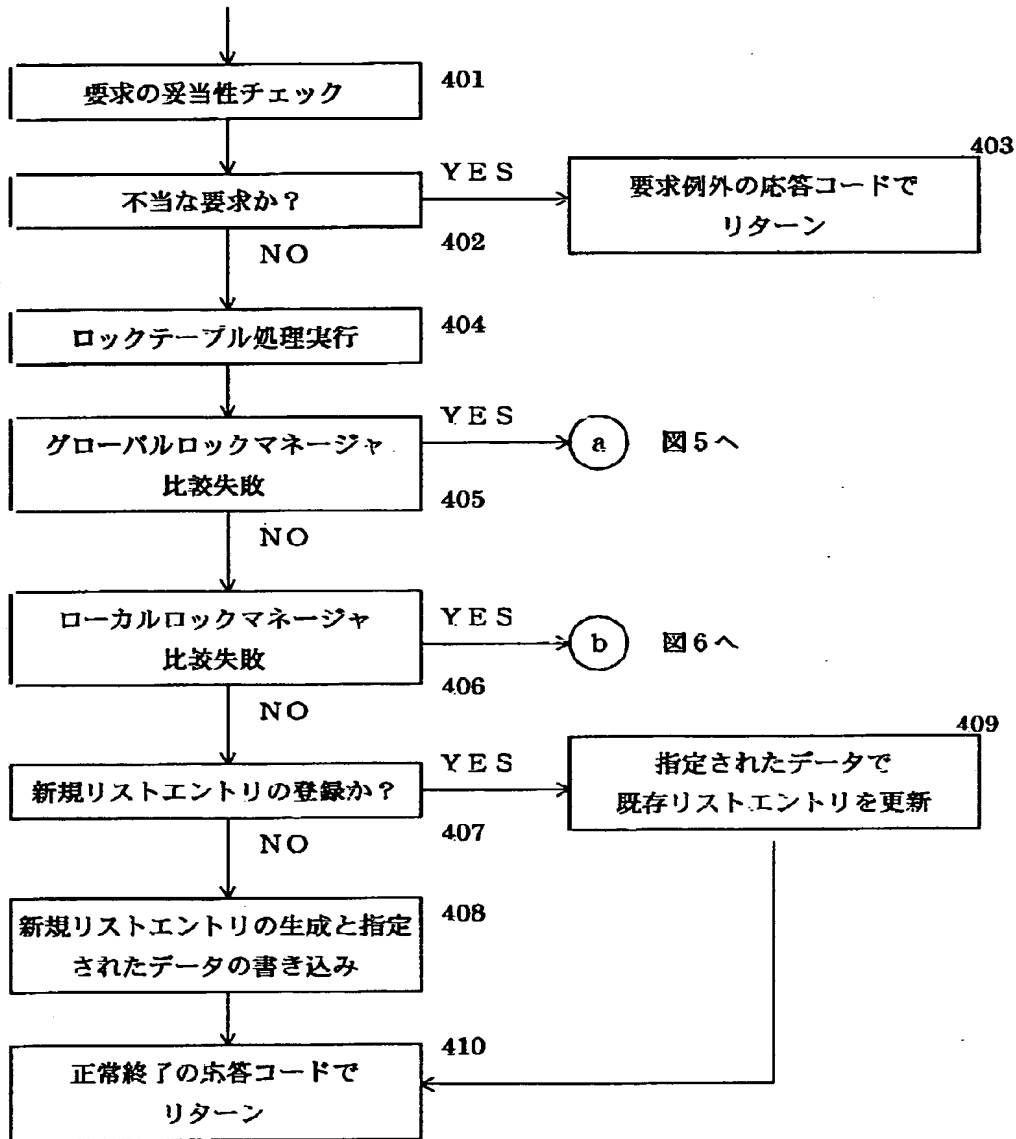


【図9】



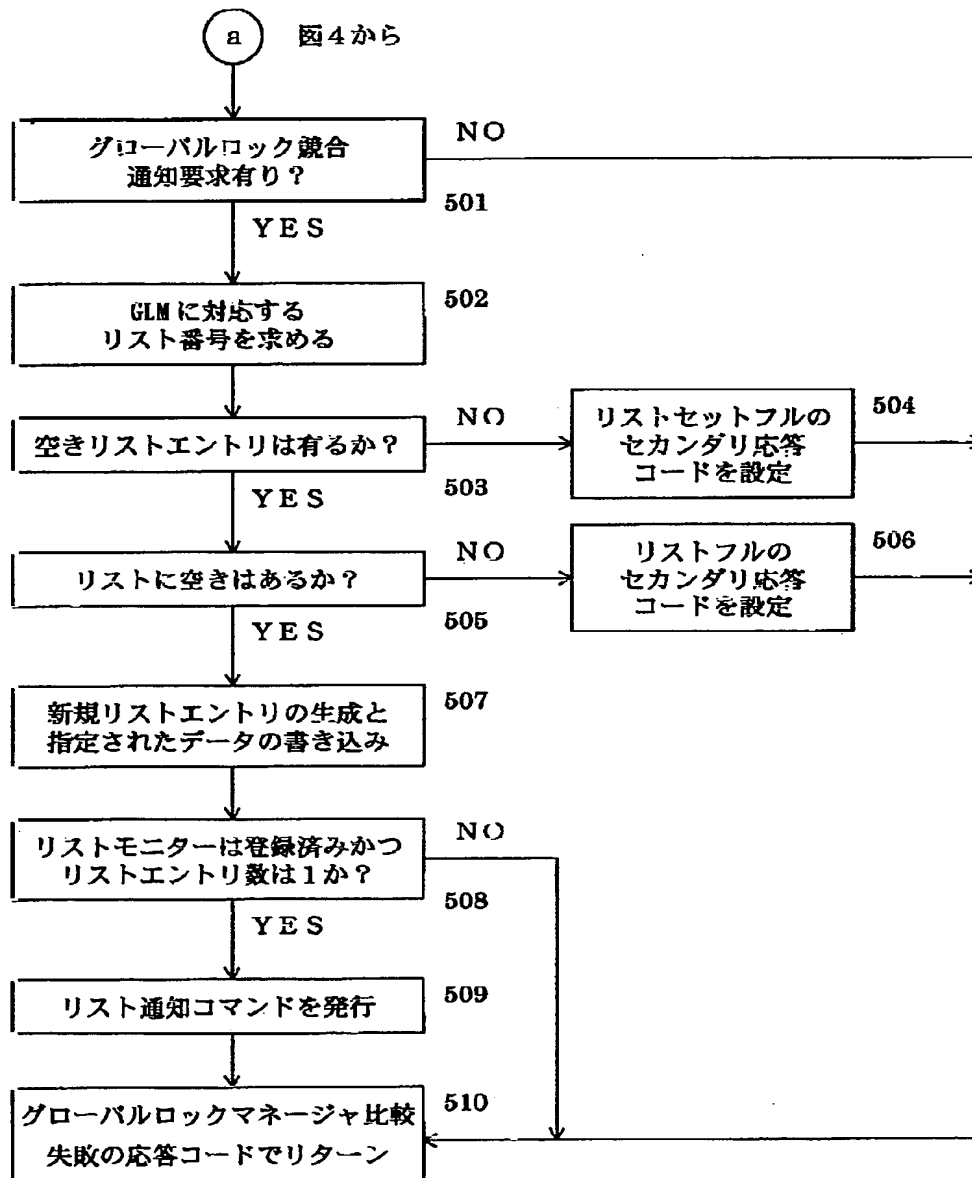
【図4】

図4



【図5】

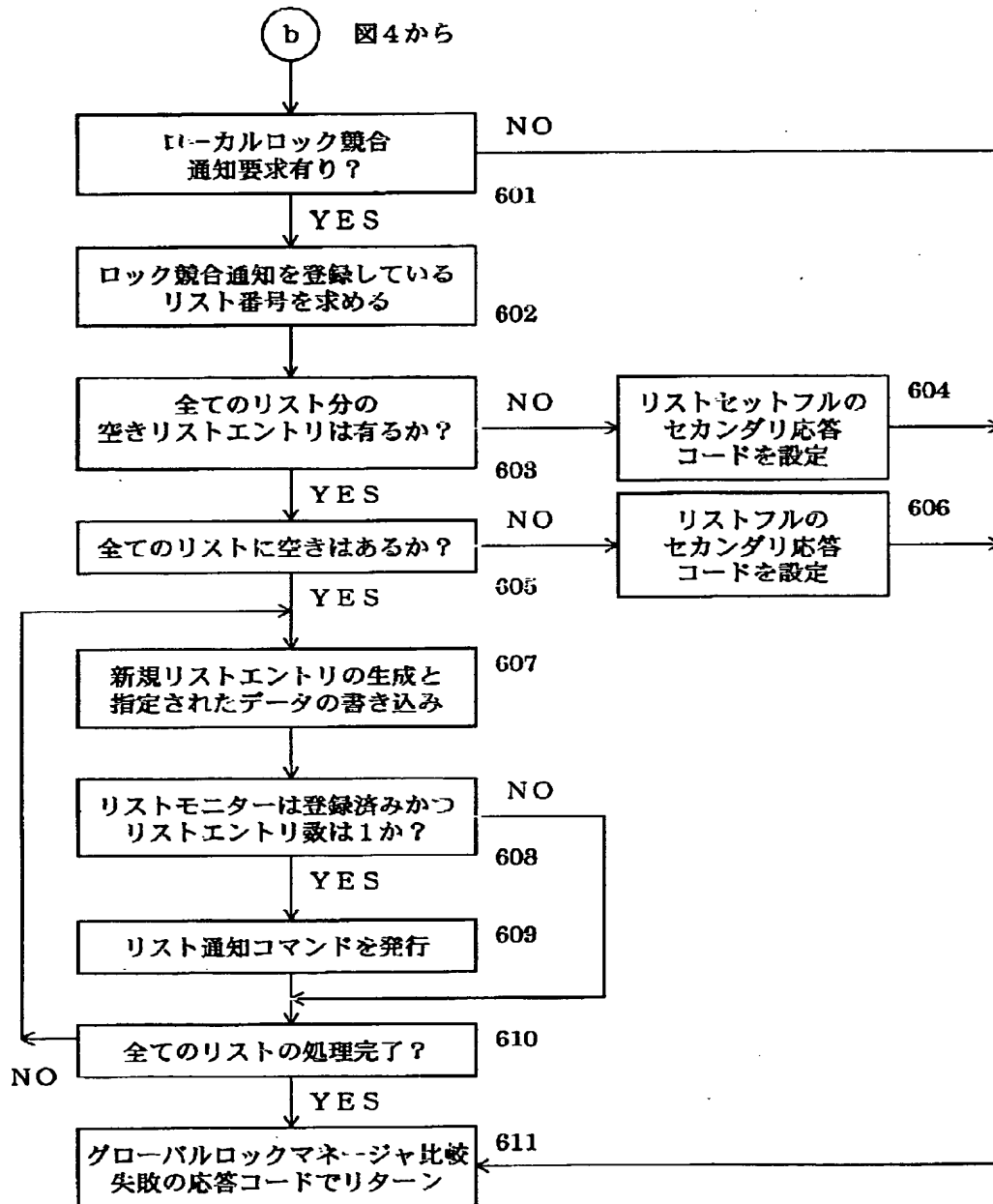
図5





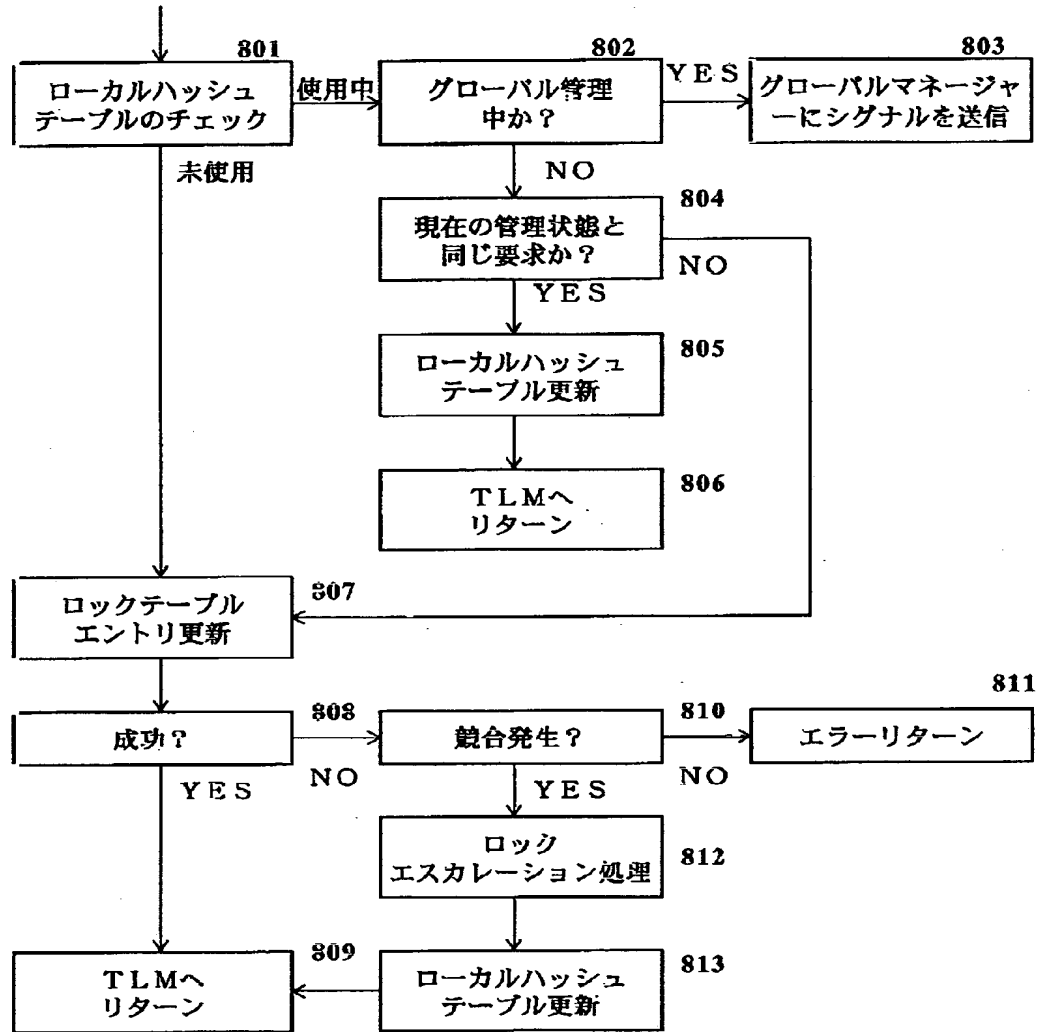
【図6】

図6



【図8】

図8



フロントページの続き

(72)発明者 横田 浩  
 神奈川県横浜市戸塚区戸塚町5030番地 株  
 式会社日立製作所ソフトウェア事業部内

(72)発明者 高山 由人  
 神奈川県横浜市戸塚区戸塚町5030番地 株  
 式会社日立製作所ソフトウェア事業部内  
 Fターム(参考) 5B045 BB31 DD18 EE11